

EVALUACIÓN CORRESPONDIENTE AL SEGUNDO PARCIAL

CORRESPONDIENTE A: (1C-2021)

FACULTAD:	Tecnología Informática		
CARRERA:	Ingeniería en Sistemas Informáticos		
ALUMNO/A:			
SEDE:	091	LOCALIZACIÓN:	
ASIGNATURA:	Programación Orientada a Objetos		
CURSO:		TURNO:	
PROFESOR:	Cardacci D.	FECHA:	
TIEMPO DE RESOLUCIÓN:	4hs	EXAMEN PARCIAL NRO:	2
MODALIDAD DE RESOLUCIÓN:	Virtual / Individual		
RESULTADOS DE APRENDIZAJE: Que el alumno: <ul style="list-style-type: none"> Conozca los conceptos de la programación OO. Identifique las clases principales en un dominio de problema. Pueda aplicar: herencia, polimorfismo, agregación, sobrecarga y sobrescritura. Elabore código de programación legible. Aplica las buenas prácticas de programación. Aplique LinQ, expresiones, lambda e interfaces. 			

Propósito:

Evaluar la capacidad del estudiante para diseñar y desarrollar un programa orientado a objetos. Su habilidad para administrar el tiempo y los recursos seleccionados para el logro del objetivo propuesto, su capacidad para integrar el marco teórico propuesto con los resultados alcanzados.

CRITERIOS DE EVALUACIÓN:

- Creatividad y originalidad de la propuesta.
- Claridad en la organización de la escritura del código.
- Precisión en el resultado obtenido.
- Utilización de los conceptos abordados en clase.
- Racionalidad y coherencia en la validación de los datos ingresados y obtenidos así como en el control de las excepciones.

El examen se considerará aprobado con una nota de 4 (cuatro) que se obtendrá con el 60% de los ítems del PROYECTO correctamente desarrollados. Es importante la originalidad del planteo.

Guía de Resolución:

Título de proyecto software:

1. Introducción

Se le solicita que desarrolle un programa orientado a objetos utilizando los conceptos abordados en la asignatura. El mismo tiene como objetivo observar cómo aplica los temas abordados en su elaboración.

2. Objetivos

Desarrollar un programa que le permita a una institución comercial administrar los cobros pendientes que deben realizarse a sus clientes

Nos informan que existen dos tipos de cobros. Los normales y los especiales. Todos se identifican por un código, poseen un nombre, fecha de vencimiento, el importe a cobrar y el cliente a quien se le cobró. De los clientes mantenemos un legajo y el nombre. Los cobros normales cuando se realizan después de la fecha de vencimiento abonan un 1% adicional por cada día de retraso y los especiales un monto fijo de \$1000 + un 2% por día de retraso. Los importes calculados por retraso se realizan al momento de recepcionar el pago y deben visualizarse discriminados además del total abonado.

No se aceptan pagos parciales. Al momento que un cliente paga (botón pagar), se le debe informar en manera discriminada el importe a pagar + el recargo si corresponde + el total en un MessageBox, aceptado este se da por pagada la deuda. Si el importe total a pagar da más de 10.000 pesos desencadenar un evento que informe sobre esta situación.

También nos informan que un cliente no puede tener pendientes más de dos importes al cobro.

Nos solicitan:

1. En la Grilla 1, poder realizar el ABM de los clientes.
2. En la Grilla 2, los cobros pendientes del cliente seleccionado en la Grilla 1. Al dar de alta un cobro siempre se asigna al cliente seleccionado en la grilla 1. Los cobros una vez dados de alta no se pueden borrar ni modificar y no pueden existir cobros que no correspondan a un cliente.
3. En la Grilla 3, los cobros cancelados del cliente seleccionado en la Grilla 1 (usar LinQ)
4. En la Grilla 4, los cobros cancelados del cliente seleccionado en la Grilla 1 ordenados por importe total de mayor a menor y viceversa. (utilizar IComparable y radioButtons para seleccionar el criterio).
5. En la Grilla 5, los cobros cancelados solo con los siguientes datos (Nombre del cliente, Importe total cancelado). (Usar LinQ y tipos anónimos)

Validar todos los datos para que no existan datos repetidos (p.e. legajos, códigos etc)

Utilizar Try ... Catch para administrar las excepciones del sistema.

No utilizar controles de tipo menú, toda la GUI debe estar en un formulario.

Observe la usabilidad (fácil de utilizar por el usuario, cantidad de clic para una operación, suma claridad en lo que el usuario debe realizar para utilizar en sistema).

3. Plan de trabajo

Desarrollar el código correspondiente al programa solicitado.

No documentar el código considerando su uso para la defensa.

Probar el código para detectar fallas que no permitan lograr los objetivos planteados.

Entregar el programa al solicitante

Forma de entrega:

Colocar en un archivo .zip^(*) el desarrollo completo y el documento del parcial (no utilice ninguna versión que no sea la estándar en .zip ni otras extensiones)

Nombre del Archivo: Sede_Asignatura_Curso_1erParcial_Apellido_Nombre.zip

Ejemplo: 091_POO_2A_1erParcial_Perez_Juan.zip